4/PRTS

Description

Method and arrangement for determining a sequence of actions for a system which has states, a transition in state between two states being performed on the basis of an action

The invention relates to a method and an arrangement for determining a sequence of actions for a system which has states, a transition in state between two states being performed on the basis of an action.

Such a method and such an arrangement are known from [1].

A financial market is described in [1] as an example for such a system which has states.

The system is described as a Markov decision problem (MDP). The structure of a system which can be described as a Markov decision problem is illustrated in **Figure 2**.

The system 201 is in a state x_t at an instant t. The state x_t can be observed by an observer of the system. On the basis of an action a_t from a set in the state x_t of possible actions, $a_t \in A(x_t)$, the system makes a transition with a certain probability into a subsequent state x_t+1 at a subsequent instant t+1.

This is illustrated diagrammatically in **Figure 2** by a loop. An observer 200 perceives 202 observable variables concerning the state x_t and takes a decision via an action 203 with which it acts on the system 201. The system 201 is usually subject to the interference 205.

Furthermore, the observer 200 obtains a gain r_t 204

$$r_t = r(x_t, a_t, x_{t+1}) \in \Re, \qquad (1)$$

which is a function of the action a_t 203 and the original state x_t at the instant t as well as of the subsequent state x_t+1 of the system at the subsequent instant t+1.

The gain r_t can assume a positive or negative skalar value, depending on whether the decision leads, with regard to a prescribable criterion, to a positive or negative system development, in [1] to an increase in capital stock or to a loss.

In a further time step, the observer 200 of the system 201 decides on the basis of the observable variables 202, 204 of the subsequent state x_{t+1} in favor of a new action a_{t+1}, etc.

A sequence of

State:	x_t	$\in$	X
Action:	a_t	$\in$	$A(x_t)$
Subsequent state:	x_t+1	$\in$	X
Gain	$r_t = r(x_t, a_t, x_{t+1})$	$\in$	$\Re$

etc. describes a trajectory of the system which is evaluated by a performance criterion which accumulates the individual gains r_t over the instants t. It is assumed by way of simplification in a Markov decision problem that the state x_t and the action a_t all contain information for the purpose of describing a transition probability $p(x_{t+1}|\cdot)$ of the system from the state x_t to the subsequent state x_{t+1}.

In formal terms, this means that:

$$p(x_{t+1}|x_t, K, x_0, a_t, K, a_0) = p(x_{t+1}|x_t, a_t). \qquad (2)$$

$p(x_{t+1}|x_t,a_t)$ denotes a transition probability for the subsequent state x_{t+1} for a given state x_t and given action a_t.

In a Markov decision problem, future states of the system 201 are thus not a function of states and actions which lie further in the past than one time step.

The characteristics of a Markov decision problem are represented below by way of summary:

X	set of possible states of the system, e.g. $X = \Re^m$,
$A(x_t)$	set of possible actions in the state
$p(x_{t+1}\|x_t,a_t)$	x_t
$r(x_t, a_t, x_{t+1})$	gain with expectation $R(x_t, a_t)$.

Starting from observable variables, the variables denoted below as training data, the aim is to determine a strategy, that is to say a sequence of functions

$$\pi = \{\mu_0, \mu_1, K, \mu_T\}, \tag{3}$$

which at each instant t map each state into an action rule, that is to say action

$$\mu_t(x_t) = a_t \tag{4}$$

Such a strategy is evaluated by an optimization function.

The optimization function specifies the expectation, the gains accumulated over time at a given strategy π, and a start state x_0.

The so-called Q-learning method is described in [1] as an example of a method of approximative dynamic programming.

An optimum evaluation function $V^*(x)$ is defined by

$$V^*(x) = \max_{\pi} V^{\pi}(x) \qquad \forall x \in X \tag{5}$$

where

$$V^{\pi}(x) = E\left[\sum_{t=0}^{\infty} \gamma^{t} r(x_t, \mu_t, x_{t+1}) | x_0 = x\right], \tag{6}$$

γ denoting a prescribable reduction factor which is formed in accordance with the following rule:

$$\gamma = \frac{1}{1+z}, \tag{7}$$

$$z \in \Re^{+}. \tag{8}$$

A Q-evaluation function $Q^*(x_t, a_t)$ is formed within the Q-learning method for each pair (state x_t, action a_t) in accordance with the following rule:

$$\begin{aligned} Q^*(x_t, a_t) := \quad & \sum_{x \in X} p(x_{t+1}|x_t, a_t) \cdot r_t + \\ & + \gamma \cdot \sum_{x \in X} p(x|x_t, a_t) \cdot \max_{a \in A}\left(Q^*(x, a)\right) \end{aligned}$$

(9)

On the basis respectively of the tupel (x_t, x_{t+1}, a_t, r_t), the Q-values Q^* (x,a) are adapted in the k+1 th iteration in accordance with the following learning rule with a prescribed learning rate η_k in accordance with the following rule:

$$Q_{k+1}(x_t, a_t) = (1 - \eta_k)Q_k(x_t, a_t) + \eta_k\left(r_t + \gamma \max_{a \in A}\left(Q_k(x_{t+1}, a)\right)\right). \quad (10)$$

Usually, the so-called Q-values $Q^*(x,a)$ are approximated for various actions a by a function approximator in each case, for example a neural network or else a polynomial classifier, with a weighting vector w^a, which contains weights of the function approximator.

A function approximator is to be understood as, for example, a neural network, a polynomial classifier or else a combination of a neural network with a polynomial classifier.

It therefore holds that:

$$Q^*(x, a) \approx Q\left(x; w^a\right). \quad (11)$$

Changes in the weights in the weighting vector w^a are based on a temporal difference d_t which is formed in accordance with the following rule:

$$d_t := r(x_t, a_t, x_{t+1}) + \gamma \max_{a \in A} Q\left(x_{t+1}; w_k^a\right) - Q\left(x_t; w_k^{a_t}\right) \quad (12)$$

The following adaptation rule for the weights of the neural network, which are included in the weighting vector w^a, follows for the Q-learning method with the use of a neural network:

$$w_{k+1}^{a_t} = w_k^{a_t} + \eta_k \cdot d_t \cdot \nabla Q\left(x_t; w_k^{a_t}\right). \qquad (13)$$

The neural network representing the system of a financial market, as described in [1], is trained using the training data which describe information on changes in prices on a financial market as time series values.

A further method of approximative dynamic programming, the so-called TD(λ)-learning method, is known from [2] and is explained in more detail in conjunction with an exemplary embodiment.

Furthermore, it is known from [3] which risk is associated with a strategy π and an initial state x_t. A method for risk avoidment is likewise known from [3].

The following optimization function, which is also referred to as an expanded Q-function $\underline{Q}^{\pi}(x_t, a_t)$, is used in the method known from [3]:

$$\text{maximize}\left(\underline{Q}^{\pi}(x_t, a_t) := r(x_t, a_t, x_{t+1}) + \inf_{\substack{x_0, x_1, K \\ p(x_0, x_1, K) > 0}} \left\{\sum_{k=1}^{\infty} \gamma^k r(x_k, \pi(x_k), x_{k+1})\right\}\right) \qquad (14)$$

The expanded Q-function $\underline{Q}^{\pi}(x_t, a_t)$ describes the worst case if the action a_t is executed in the state x_t and the strategy π is followed thereupon.

The optimization function $\underline{Q}^{\pi}(x_t, a_t)$ for

$$\underline{Q}^{*}(x_t, a_t) := \max_{\pi \in \Pi} \underline{Q}^{\pi}(x_t, a_t)$$

(15)

,

is given by the following rule:

$$\underline{Q}^{*}(x_t, a_t) = \min_{\substack{x \in X \\ p(x_{t+1}|x_t, a_t) > 0}} \left(r(x_t, a_t, x) + \gamma \cdot \max_{a \in A} \underline{Q}^{*}(x, a) \right). \quad (16)$$

A substantial disadvantage of this mode of procedure is to be seen in that only the worst case is taken into account when finding the strategy. However, this reflects the requirements of the most varied technical systems only to an inadequate extent.

Furthermore, it is known from [4] to formulate access control for a communications network and the routing within the communications network as a problem of dynamic programming.

The invention is therefore based on the problem of specifying a method and an arrangement for determining a sequence of actions for a system, in which method or action an increased flexibility in determining the strategy is achieved.

The problem is solved by the method and by the arrangement in accordance with the features of the independent patent claims.

In a method for computer-aided determination of a sequence of actions for a system which has states, a transition in state between two states being performed on the basis of an action, the determination of the sequence of actions is performed in such a way that a sequence of states resulting from the sequence of actions is optimized with regard to a prescribed optimization function, the optimization function including a variable parameter with the aid of which it is possible to set a risk which the resulting sequence of states has with respect to a prescribed state of the system.

An arrangement for determining a sequence of actions for a system which has states, a transition in state between two states being performed on the basis of an action, has a processor which is set up in such a way that the determination of the sequence of actions can be performed in such a way that a sequence of states resulting from the sequence of actions is optimized with regard to a prescribed optimization function, the optimization function including a variable parameter with the aid of which it is possible to set a risk which the resulting sequence of states has with respect to a prescribed state of the system.

It becomes possible for the first time owing to the invention to specify a method for determining a sequence of actions at a freely prescribable level of accuracy when finding a strategy for a possible closed-loop control or open-loop control of the system, in general for influencing it.

Preferred developments of the invention follow from the dependent claims.

The developments described below are valid both for the method and for the arrangement, the processor being respectively set up in the development of arrangement in such a way that the development can be implemented.

In a preferred refinement, a method of approximative dynamic programming is used for the purpose of determination, for example a method based on Q-learning or else a method based on TD(λ)-learning.

Within Q-learning, the optimization function OFQ is preferably formed in accordance with the following rule:

$$OFQ = Q\left(x; w^{a}\right),$$

- x denoting a state in a state space X
- a denoting an action from an action space A, and
- w^{a} denoting the weights of a function approximator which belong to the action a.

The following adaptation step is executed during Q-learning in order to determine the optimum weights w^{a} of the function approximator:

$$w_{t+1}^{a_t} = w_{t}^{a_t} + \eta_t \cdot \aleph^{\kappa}\left(d_t\right) \cdot \nabla Q\left(x_t; w_{t}^{a_t}\right)$$

with the abbreviation

$$d_t = r\left(x_t, a_t, x_{t+1}\right) + \gamma \max_{a \in A} Q\left(x_{t+1}, w_{t}^{a}\right) - Q\left(x_t, w_{t}^{a_t}\right)$$

- x_t, x_t+1 respectively denoting a state in the state space X,
- a_t denoting an action from an action space A,
- γ denoting a prescribable reduction factor,
- w_{t}^{at} denoting the weighting vector associated with the action a_t before the adaptation step,
- w_{t+1}^{at} denoting the weighing vector associated with the action a_t after the adaptation step,
- η_t (t = 1, ...) denoting a prescribable step size sequence,

- $\kappa \in [-1; 1]$ denoting a risk monitoring parameter,
- $\aleph^{\kappa}$ denoting a risk monitoring function $\aleph^{\kappa}(\xi) = (1 - \kappa\mathrm{sign}(\xi))\xi$,
- $\nabla Q(\cdot;\cdot)$ denoting the derivation of the function approximator according to its weights, and
- $r(x_t, a_t, x_{t+1})$ denoting a gain upon the transition of state from the state x_t to the subsequent state x_{t+1}.

The optimization function is preferably formed in accordance with the following rule within the TD(λ)-learning method:

$$OFTD = J(x;w)$$

- x denoting a state in a state space X,
- a denoting an action from an action space A, and
- w denoting the weights of a function approximator.

The following adaptation step is executed during TD(λ)-learning in order to determine the optimum weights w of the function approximator:

$$w_{t+1} = w_t + \eta_t \cdot \aleph^{\kappa}(d_t) \cdot z_t$$

with the abbreviations

$$d_t = r(w_t, a_t, x_{t+1}) + \gamma J(x_{t+1}; w_t) - J(x_t; w_t),$$

$$z_t = \lambda \cdot \gamma \cdot z_{t-1} + \nabla J(x_t; w_t),$$

$$z_{-1} = 0$$

- x_t, x_{t+1} respectively denoting a state in the state space X,
- a_t denoting an action from an action space A,
- γ denoting a prescribable reduction factor,
- w_t denoting the weighting vector before the adaptation step,
- w_{t+1} denoting the weighting vector after the adaptation step,
- η_t $(t = 1, \ldots)$ denoting a prescribable step size sequence,
- $\kappa \in [-1; 1]$ denoting a risk monitoring parameter,
- $\aleph^{\kappa}$ denoting a risk monitoring function $\aleph^{\kappa}(\xi) = (1 - \kappa \operatorname{sign}(\xi))\xi$,
- $\nabla J(\cdot;\cdot)$ denoting the derivation of the function approximator according to its weights, and
- $r(x_t, a_t, x_{t+1})$ denoting a gain upon the transition of state from the state x_t to the subsequent state x_{t+1}.

The system is preferably a technical system of which before the determination measured values are measured which are used in determining the sequence of actions.

The technical system can be subjected to open-loop control or else closed-loop control with the use of the determined sequence of actions.

The system is preferably modeled as a Markov decision problem.

The method or the arrangement is preferably used in a traffic management system or in a communications system, the sequence of actions being used in a communications network to carry out access control or a routing, that is to say a path allocation.

Furthermore, the system can be a financial market which is modeled by a Markov decision problem, the change in the financial market, for example the change in an index of stocks or else a rate of exchange on a foreign exchange market being analyzed by using the method and/or the arrangement and it being possible to intervene in the market in accordance with the sequence of determined actions.

Exemplary embodiments of the invention are illustrated in the figures and explained in more detail below.

Figure 1 shows a flowchart in which individual method steps of the first exemplary embodiment are illustrated;

Figure 2 shows a sketch of a system which can be modeled as a Markov decision problem;

Figure 3 shows a sketch of a communications network in which access control is carried out in a switching unit;

Figure 4 shows a symbolic sketch of a function approximator with the aid of which a method of approximative dynamic programming is implemented;

Figure 5 shows a further sketch of a plurality of function approximators with the aid of which approximative dynamic programming is implemented; and

Figure 6 shows a sketch of a traffic management system which is subjected to closed-loop control in accordance with an exemplary embodiment.

First exemplary embodiment: access control and routing.

Figure 3 shows a communications network 300, which has a multiplicity of switching units 301a, 301b, ..., 301i, ... 301n, which are interconnected via connections 302a, 302b, 302j, ... 302m.

Furthermore, a first terminal 303 is connected to a first switching unit 301a. From the first terminal 303, the first switching unit 301a is sent a request message 304 which requests preservation of a prescribed bandwidth within the communications network 300 for the purpose of transmitting data (video data, text data).

It is determined in the first switching unit 301a in accordance with a strategy described below whether the requested bandwidth is available in the communications network 300 on a specified, requested connection (step 305).

The request is refused (step 306) if this is not the case.

If sufficient bandwidth is available, it is checked in a further checking step (step 307) whether the bandwidth can be reserved.

The request is refused (step 308) if this is not the case.

Otherwise, the first switching unit 301a selects a route from the first switching unit 301a via further switching units 301i to a second terminal 309 with which the first terminal 303 wishes to communicate, and a connection is initialized (step 310).

The starting point below is a communications network 300 which comprises a set of switching units

$$N = \{1, K, n, K, N\} \quad (17)$$

and a set of physical connections

$$L = \{1, K, 1, K, L\}, \quad (18)$$

a physical connection 1 having a capacity of B(1) bandwidth units.

A set

$$M = \{1, K, m, K, M\} \quad (19)$$

of different types of service m are available, a type of service m being characterized by

- a bandwidth requirement b(m),
- an average connection time $\frac{1}{V(m)}$ and
- a gain c(m) which is obtained whenever a call request of the corresponding type of service m is accepted.

The gain c(m) is given by the amount of money which a network operator of the communications network 300 bills a subscriber for a connection of the type of service. Clearly, the gain c(m) reflects different priorities, which can be prescribed by the network operator and which he associates with different services.

A physical connection 1 can simultaneously provide any desired combination of communications connections as long as the bandwidth used for the communications connections does not exceed the bandwidth available overall for the physical connection.

If a new communications connection of type m is requested between a first node i and a second node j (terminals are also denoted as nodes), the requested communications connection can, as represented above, either be accepted or be refused.

If the communications connection is accepted, a route is selected from a set of prescribed routes. This selection is denoted as a routing. b(m) bandwidth units are used in the communications connection of type m for each physical connection along the selected route for the duration of the connection.

Thus, during access control (call admission control), a route can be selected within the communications network 300 only when the selected route has sufficient bandwidth available.

The aim of the access control and of the routing is to maximize a long term gain which is obtained by acceptance of the requested connections.

At an instant t, the technical system which is the communications network 300 is in a state x_t which is described by a list of routes via existing connections, by means of which lists it is shown how many connections of which type of service are using the respective routes at the instant t.

Events w, by means of which a state x_t could be transferred into a subsequent state x_{t+1}, are the arrival of new connection request messages, or else the termination of a connection existing in the communications network 300.

In this exemplary embodiment, an action a_t at an instant t owing to a connection request is the decision as to whether a connection request is to be accepted or refused and, if the connection is accepted, the selection of the route through the communications network 300.

The aim is to determine a sequence of actions, that is to say clearly to determine the learning of a strategy with actions relating to a state x_t in such a way that the following rule is maximized:

$$E\left(\sum_{k=0}^{\infty} e^{-\beta t_k} \cdot g\left(x_{t_k}, \omega_k, a_{t_k}\right)\right), \qquad (20)$$

- E{.} denoting an expectation,
- t_k denoting an instant at which a kth event takes place,
- $g\left(x_{t_k},\omega_k,a_{t_k}\right)$. denoting the gain which is associated with the kth event, and
- β denoting a reduction factor which evaluates an immediate gain as being more valuable than a gain at instants lying further in the future.

Different implementations of a strategy lead normally to different overall gains G:

$$G = \sum_{k=0}^{\infty} e^{-\beta t_k} \cdot g\left(x_{t_k}, \omega_k, a_{t_k}\right). \qquad (21)$$

The aim is to maximize the expectation of the overall gain G in accordance with the following rule J:

$$J = E\left\{\sum_{k=0}^{\infty} e^{-\beta t_k} \cdot g\left(x_{t_k}, \omega_k, a_{t_k}\right)\right\}, \qquad (22)$$

it being possible to set a risk which reduces the overall gain G of a specific implementation of access control and of a routing strategy to below the expectation.

The TD(λ)-learning method is used to carry out the access control and the routing.

The following target function is used in this exemplary embodiment:

$$J^*(x_t) = E_\tau\left\{e^{-\beta\tau}\right\}E_\omega\left\{\max_{a \in A}\left[g(x_t, \omega_t, a) + J^*(x_{t+1})\right]\right\}, \qquad (23)$$

- A denoting an action space with a prescribed number of actions which are respectively available in a state x_t,
- τ denoting a first instant at which a first event ω occurs, and
- x_{t+1} denoting a subsequent state of the system.

An approximated value of the target value $J^*(x_t)$ is learned and stored by employing a function approximator 400 (compare **Figure 4**) with the use of training data.

Training data are data previously measured in the communications network 300 and relating to the behavior of the communications network 300 in the case of incoming connection requests 304 and of termination of messages. This time sequence of states is stored, and these training data are used to train the function approximator 400 in accordance with the learning method described below.

A number of connections of in each case one type of service m on a route of the communications network 300 serve in each case as input variable of the function approximator 400 for each input 401, 402, 403 of the function approximator 400. These are represented symbolically in **Figure 4** by blocks 404, 405, 406.

An approximated target value $\tilde{J}$ of the target value J^* is the output variable of the function approximator 400.

Figure 5 shows a detailed representation of the function approximator 500, which in this case has several component function approximators 510, 520 of the function approximator 500. One output variable is the approximated target value $\tilde{J}$, which is formed in accordance with the following rule:

$$\tilde{J}(x_t, \Theta) = \sum_{l=1}^{L} \tilde{J}^{(l)}\left(x_t^{(l)}, \Theta_t^{(l)}\right). \tag{24}$$

The input variables of the component function approximators 510, 520, which are present at the inputs 511, 512, 513 of the first component function approximator 510, or at the inputs 521, 522 and 523 of the second component function approximator 520 are, in turn, respectively a number of types of service of a type m in a physical connection r in each case, symbolized by blocks 514, 515, 516 for the first component function approximator, and 524, 525 and 526 for the second component function approximator 520.

Component output variables 530, 531, 532, 533 are fed to an adder unit 540, and the approximated target variable $\tilde{J}$ is formed as output variable of the adder unit.

Let it be assumed that the communications network 300 is in the state x_{tk} and that a request message with which a type of service m of class m is requested for a connection between two nodes i, j reaches the first switching unit 301a.

A list of permitted routes between the nodes i and j is denoted by R(i, j), and a list of all possible routes is denoted by

$$\tilde{R}\left(i, j, x_{t_k}\right) \subset R(i, j) \qquad (25)$$

as a subset of the routes R(i, j) which could implement a possible connection with regard to the available and requested bandwidth.

For each possible route r, $r \in \tilde{R}\left(i, j, x_{t_k}\right)$, a subsequent state $x_{t_k+1}\left(x_{t_k}, \omega_k, r\right)$ is determined which results from the fact that the connection request 304 is accepted and the connection on the route r is made available to the requesting first terminal 303.

This is illustrated in **Figure 1** as second step (step 102), the state of the system and the respective event being respectively determined in a first step (step 101).

A route r* to be selected is determined in a third step (step 103) in accordance with the following rule:

$$r^* = \arg \max_{r \in \tilde{R}\left(i, j, x_{t_k}\right)} \tilde{J}\left(x_{t_k+1}\left(x_{t_k}, \omega_k, r\right), \Theta_t\right). \qquad (26)$$

A check is made in a further step (step 104) as to whether the following rule is fulfilled:

$$c(m) + \tilde{J}\left(x_{t_k+1}\left(x_{t_k}, \omega_k, r^*\right), \Theta_t\right) < \tilde{J}\left(x_{t_k}, \Theta_t\right). \qquad (27)$$

If this is the case, the connection request 304 is rejected (step 105), otherwise the connection is accepted and "switched through" to the node j along the selected route r* (step 106).

Weights of the function approximator 400, 500 which are adapted in the TD(λ)-learning method to the training data, are stored in a parameter vector θ for an instant t in each case, such that an optimized access control and an optimized routing are achieved.

During the training phase, the weighting parameters are adapted to the training data applied to the function approximator.

A risk parameter κ is defined with the aid of which a desired risk, which the system has with regard to a prescribed state owing to a sequence of actions and states, can be set in accordance with the following rules:

$-1 \leq \kappa < 0$:	risky learning,
$\kappa = 0$:	neutral learning with regard to the risk,
$0 < \kappa < 1$:	risk-avoiding learning,
$\kappa = 1$:	worst-case learning.

Furthermore, a prescribable parameter $0 \leq \lambda \leq 1$ and a step size sequence γ_k are prescribed in the learning method.

The weighting values of the weighting vector Θ are adapted to the training data on the basis of each event ω_{t_k} in accordance with the following adaptation rule:

$$\Theta_k = \Theta_{k-1} + \gamma_k \aleph^{\kappa}(d_k) z_t, \qquad (28)$$

in which case

$$d_k = e^{-\beta(t_k - t_{k-1})}\left(g\left(x_{t_k}, \omega_k, a_{t_k}\right) + \tilde{J}\left(x_{t_k}, \Theta_{k-1}\right)\right) - \tilde{J}\left(x_{t_{k-1}}, \Theta_{k-1}\right) \tag{29}$$

$$z_t = \lambda e^{-\beta(t_{k-1} - t_{k-2})} z_{t-1} + \nabla_{\Theta}\tilde{J}\left(x_{t_{k-1}}, \Theta_{k-1}\right), \tag{30}$$

and

$$\aleph^{\kappa}(\xi) = \left(1 - \kappa\,\mathrm{sign}(\xi)\right)\xi\,. \tag{31}$$

It is assumed that: $z_{-1} = 0$.

The function

$$g\left(x_{t_k}, \omega_k, a_{t_k}\right) \tag{32}$$

denotes the immediate gain in accordance with the following rule:

$$g\left(x_{t_k}, \omega_k, a_{t_k}\right) = \begin{cases} c(m) & \text{when } \omega_{t_k} \text{ is a service request for a type of service m, and the connection is accepted} \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

Thus, as described above, a sequence of actions is determined with regard to a connection request such that a connection request is either rejected or accepted on the basis of an action. The determination is performed taking account of an optimization function in which the risk can be set by means of a risk control parameter $\kappa \in [-1; 1]$ in a variable fashion.

Second exemplary embodiment: Traffic management system

Figure 6 shows a road 600 on which automobiles 601, 602, 603, 604, 605 and 606 are being driven.

Conductor loops 610, 611 integrated into the road 600 receive electric signals in a known way and feed the electric signals 615, 616 to a computer 620 via an input/output interface 621. In an analog-to-digital converter 622 connected to the input/output interface 621, the electric signals are digitized into a time series and stored in a memory 623, which is connected by a bus 624 to the analog-to-digital converter 622 and a processor 625. Via the input/output interface 621, a traffic management system 650 is fed control signals 651 from which it is possible to set a prescribed speed stipulation 652 in the traffic management system 650, or else further particulars of traffic regulations, which are displayed via the traffic management system 650 to drivers of the vehicles 601, 602, 603, 604, 605 and 606.

The following local state variables are used in this case for the purpose of traffic modeling:

- traffic flow rate v,
- vehicle density ρ (ρ = number of vehicles per kilometer $\frac{\mathrm{Fz}}{\mathrm{km}}$).
- traffic flow q (q = number of vehicles per hour $\frac{\mathrm{Fz}}{\mathrm{h}}$, ($q = v * \rho$)), and
- speed restrictions 652 displayed by the traffic management system 650 at an instant in each case.

The local state variables are measured as described above by using the conductor loops 610, 611.

These variables $(v(t), \rho(t), q(t))$ therefore represent a state of the technical system of "traffic" at a specific instant t.

In this exemplary embodiment, the system is therefore a traffic system which is controlled by using the traffic management system 650.

In this second exemplary embodiment, an extended Q-learning method is described as method of approximative dynamic programming.

The state x_t is described by a state vector

$$x(t) = \left(v(t), \rho(t), q(t)\right). \tag{34}$$

The action a_t denotes the speed restriction 652, which is displayed at the instant t by the traffic management system 650.

The gain $r(x_t, a_t, x_{t+1})$ describes the quality of the traffic flow which was measured between the instants t and t+1 by the conductor loops 610 and 611. In this second exemplary embodiment, $r(x_t, a_t, x_{t+1})$ denotes

- the average speed of the vehicles in the time interval [t, t + 1]

or

- the number of vehicles which have passed the conductor loops 610 and 611 in the time interval [t, t + 1]

or

- the variance of the vehicle speeds in the time interval [t, t + 1], or

- a weighted sum from the above variables.

A value of the optimization function OFQ is determined for each possible action a_t, that is to say for each speed restriction which can be displayed by the traffic management system 650, an estimated value of the optimization function OFQ being realized in each case as a neural network.

This results in a set of evaluation variables for the various actions a_t in the system state x_t.

Those actions a_t for which the maximum evaluation variable OFQ has been determined in the current system state x_t are selected in a control phase from the possible actions a_t, that is to say from the set of the speed restrictions which can be displayed by the traffic management system 650.

In accordance with this exemplary embodiment, the adaptation rule, known from the Q-learning method, for calculating the optimization function OFQ is extended by a risk control function $\aleph^{\kappa}(.)$, which takes account of the risk.

In turn, the risk control parameter κ is prescribed in accordance with the strategy from the first exemplary embodiment in the interval of $[-1 \le \kappa \le 1]$, and represents the risk which a user wishes to run in the application with regard to the control strategy to be determined.

The following evaluation function OFQ is used in accordance with this exemplary embodiment:

$$\mathrm{OFQ} = Q\left(x; w^{a}\right), \qquad (35)$$

- $x = (v; \rho; q)$ denoting a state of the traffic system,
- a denoting a speed restriction from the action space A of all speed restrictions which can be displayed by the traffic management system 650, and
- w^a denoting the weights of the neural network which belong to the speed restriction a.

The following adaptation step is executed in Q-learning in order to determine the optimum weights w^a of the neural network:

$$w_{t+1}^{a_t} = w_t^{a_t} + \eta_t \cdot \aleph^{\kappa}\left(d_t\right) \cdot \nabla Q\left(x_t; w_t^{a_t}\right) \qquad (36)$$

using the abbreviation

$$d_t = r\left(x_t, a_t, x_{t+1}\right) + \gamma \max_{a \in A} Q\left(x_{t+1}, w_t^{a}\right) - Q\left(x_t, w_t^{a_t}\right) \qquad (37)$$

- x_t, x_{t+1} denoting in each case a state of the traffic system in accordance with rule (34),
- a_t denoting an action, that is to say a speed restriction which can be displayed by the traffic management system 650,
- γ denoting a prescribable reduction factor,
- $w_t^{a_t}$ denoting the weighting vector belonging to the action a_t, before the adaptation step,
- $w_{t+1}^{a_t}$ denoting the weighting vector belonging to the action a_t, after the adaptation step,
- η_t $(t = 1, \ldots)$ denoting a prescribable step size sequence,

- $\kappa \in [-1; 1]$ denoting a risk control parameter,
- $\aleph^{\kappa}$ denoting a risk control function $\aleph^{\kappa}(\xi) = (1 - \kappa \mathrm{sign}(\xi))\xi$,
- $\nabla_Q(\cdot;\cdot)$ denoting the derivative of the neural network with respect to its weights, and
- $r(x_t, a_t, x_{t+1})$ denoting a gain upon the transition in state from the state x_t to the subsequent state x_{t+1}.

An action a_t can be selected at random from the possible actions a_t during learning. It is not necessary in this case to select the action a_t which has led to the largest evaluation variable.

The adaptation of the weights has to be performed in such a way that not only is a traffic control achieved which is optimized in terms of the expectation of the optimization function, but that also account is taken of a variance of the control results.

This is particularly advantageous since the state vector x(t) models the actual system of traffic only inadequately in some aspects, and so unexpected disturbances can thereby occur. Thus, the dynamics of the traffic, and therefore of its modeling, depend on further factors such as weather, proportion of trucks on the road, proportion of mobile homes, etc., which are not always integrated in the measured variables of the state vector x(t). In addition, it is not always ensured that the road users immediately implement the new speed instructions in accordance with the traffic management system.

A control phase on the real system in accordance with the traffic management system takes place in accordance with the following steps:

1. The state x_t is measured at the instant t at various points in the traffic system of traffic and yields a state vector $x(t) := (v(t), \rho(t), q(t))$.

2. A value of the optimization function is determined for all possible actions a_t, and that action a_t with the highest evaluation in the optimization function is selected.

The following publications are cited in this document:

[1] R, Neuneier, Enhancing Q-Learning for Optimal Asset Allocation, Proceedings of the Neural Information Processing Systems, NIPS 1997

[2] R.S. Sutton, Learning to predict by the method of temporal differences, Machine Learning, 3:9-44, 1988

[3] M. Heger, Risk and Reinforcement Learning: Concepts and Dynamic Programming, ZKW Bericht Nr. 8/94, Zentrum für Kognitionswissenschaften [Center for Cognitive Sciences], Bremen University, ISSN 0947-0204, December 1994

[4] D.P. Bertsekas, Dynamic Programming and Optimal Control, Athena Scientific, Belmont, MA, 1995